photonics

MDPI

Article

The Stability Optimization of Indoor Visible 3D Positioning Algorithms Based on Single-Light Imaging Using Attention Mechanism Convolutional Neural Networks

Wenjie Ji [1], Lianxin Hu [1], Xun Zhang [1,2,*], Jiongnan Lou [1], Hongda Chen [1] and Zefeng Wang [1]

[1] School of Information Engineering, Huzhou University, Huzhou 313000, China; 2022388109@stu.zjhu.edu.cn (W.J.); 03080@zjhu.edu.cn (L.H.); 03261@zjhu.edu.cn (H.C.); zefeng.wang@zjhu.edu.cn (Z.W.)
[2] Institut Supérieur d'Électronique de Paris, LISIT-ECoS, 75020 Paris, France
* Correspondence: xun.zhang@isep.fr

Abstract: In recent years, visible light positioning (VLP) techniques have been gaining popularity in research. Among them, the scheme of using a camera as a receiver provides a low-cost, high-precision positioning capability and easy integration with existing multimedia devices and robots. However, the pose changes of the receiver can lead to image distortion and light displacement, significantly increasing positioning errors. Addressing these errors is crucial for enhancing the accuracy of VLP. Most current solutions rely on gyroscopes or Inertial Measurement Units (IMUs) for error optimization, but these approaches often add complexity and cost to the system. To overcome these limitations, we propose a 3D positioning algorithm based on an attention mechanism convolutional neural network (CNN) aimed at reducing the errors caused by angles. We designed experiments and comparisons within a rotation angle range of ±15 degrees. The results demonstrate that the average error for 2D positioning is within 5.74 cm and the height error is within 3.92 cm, and the average error for 3D positioning is within 6.8 cm. Among the four groups of experiments for 3D positioning, compared to the traditional algorithm, the improvements were 7.931 cm, 15.569 cm, 6.004 cm, and 16.506 cm. The experiments indicate that it can be applied to high-precision visible light positioning for single-light imaging.

Keywords: CNN; VLC; VLP; OCC; indoor positioning; attention mechanism

check for updates

Citation: Ji, W.; Hu, L.; Zhang, X.; Lou, J.; Chen, H.; Wang, Z. The Stability Optimization of Indoor Visible 3D Positioning Algorithms Based on Single-Light Imaging Using Attention Mechanism Convolutional Neural Networks. *Photonics* **2024**, *11*, 794. https://doi.org/10.3390/photonics11090794

Received: 17 July 2024
Revised: 17 August 2024
Accepted: 21 August 2024
Published: 26 August 2024

1. Introduction

Over the years, positioning technology has played a crucial role in various fields such as transportation, production, navigation, and daily life. In outdoor settings, wireless signals can be transmitted in open spaces, a challenge often addressed using technologies like the Global Positioning System (GPS). However, the performance of a GPS for indoors is significantly degraded due to transmission hindrance. With indoor positioning widely applied in spaces such as shopping malls, museums, warehouses, and parking lots, the demand for low-cost, high-precision positioning technology has rapidly increased [1]. Currently, indoor positioning methods include Wireless Fidelity (Wi-Fi), Zigbee, Bluetooth, Ultra-Wideband (UWB), Infrared (IR), ultrasound, and Radio-Frequency Identification (RFID) [2]. With the widespread adoption of LED lighting in recent years and considering factors such as infrastructure coverage, cost, positioning accuracy, information security, and abundant spectrum resources, VLP has emerged as a research hotspot [3].

Visible light positioning systems require either multiple lights or a single light to be realized. Work [4–9] proposed positioning algorithms based on multiple lights, inferring position information by leveraging the geometric relationships or deformations between lights to achieve indoor positioning. However, the multiple light positioning system is demanding in terms of layout and lacks robustness and flexibility. Some of the works also discuss single-light positioning, but it does not guarantee high precision or requires high-cost receiving devices. In [10,11], a positioning method based on a single LED with a beacon was proposed, utilizing the fundamental principle of geometric relationships among multiple lights for auxiliary positioning. Work [12,13] proposed a method using binocular cameras and a single light to obtain the position, but it incurs high application costs and loses the good adaptability of monocular camera applications. We proposed a trilateral positioning method using a single rectangular light in previous work [14].

During positioning, the unavoidable random angles of the receiver can cause errors in the results. In work [2,3,15–17], algorithms relying on a single light and gyroscopes or IMU sensors for positioning were proposed. Work [16] proposed a single-light positioning method using PD, camera, and gyroscope. It used a hybrid RSS/AOA-based algorithm to achieve 3D positioning. In work [18], the gyroscope angle information is used to reconstruct the image and reduce the positioning error caused by the change of camera pose. However, it only tested for an angle of five. In practical application scenarios, the receiver often encounters significant angles. In summary, most of the single-light positioning methods rely heavily on sensors to handle the smaller angular changes at the receiver.

So, based on the above analysis, we propose a high-precision 3D positioning algorithm based on single-light imaging. The main contributions of this study are as follows:

1. We use the multi-head attention mechanism (MHA) and residual convolutional neural network (resnet50) to form a new model MHA-Resnet50, which effectively avoids model overfitting and makes training more efficient.
2. The dependencies between image features and pose are automatically learned by the model, and the predicted coordinates are regressed.
3. The reduced use of IMU sensors simplifies the algorithm and enhances the robustness of the positioning system.

Accurate positioning at a ±15 angle is achieved with a low-resolution image of 1280 × 720. The results are better compared to the proposed methods of works [18,19].

The rest of this paper is organized as follows. Section 2 briefly describes the system and analyzes the issues that need to be addressed. Section 3 introduces the proposed MHA-Resnet50. Section 4 describes the experimental environment, and how the model is trained. The positioning results of the model are also analyzed in comparison with the original algorithm. Finally, Section 5 concludes the paper.

2. Visible Light Positioning System and Issue Analysis

2.1. System Overview

Figure 1 illustrates an indoor VLP system utilizing a single LED. The transmitter consists of a rectangular LED and a signal modulator mounted on the ceiling parallel to the floor. Each LED is assigned a unique ID associated with its actual spatial position, and these correspondences are stored in a database. After modulation processing, the LED repetitively transmits its ID. The receiver side is a camera connected to a computer for capturing video frames. Through image frame processing, the LED ID can be decoded and the corresponding LED world coordinates will be identified. At the same time, the 3D relative coordinate position of the camera with respect to the LED can be calculated by MHA-Resnet50. The specific modeling method and prediction process of MHA-Resnet50 are illustrated in Section 3. Next, the spatial coordinate position of the camera can be calculated by combining the world coordinates of the LED.

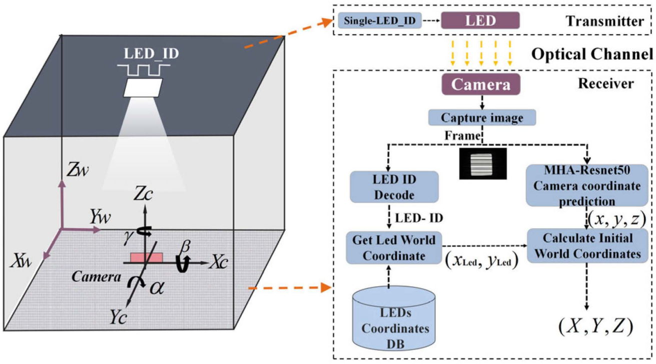


Figure 1. Visible light positioning system.

2.2. Foundation

In order to conduct the analysis easily, in this paper, the complex nonlinear model of the camera lens system is simplified to a simple pinhole camera model. As shown in Figure 2, the relationship between the world coordinate system (X_w, Y_w, Z_w) and the pixel coordinate system (u, v) can be expressed as (1):

$$Z_C \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f}{dx} & 0 & u_0 & 0 \\ 0 & \frac{f}{dy} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \tag{1}$$

where Zc represents the Z coordinate of the camera in the world coordinate system, (X_w, Y_w, Z_w) is its actual position in space, and u and v are the coordinates of the point on the image. Focal length f and physical dimensions d_x, d_y of the pixels constitute the intrinsic matrix that connects the pixel coordinate system to the camera coordinate system [14,20]. Similarly, translation vector T and rotation matrix R constitute the extrinsic matrix that connects the camera to the world coordinate system. R is given by (2).

$$\begin{aligned} R &= R_X(\alpha)R_Y(\beta)R_Z(\gamma) \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{2}$$

α, β, γ denote the pitch, roll, and azimuth.

The original positioning algorithm is implemented based on the imaging principles shown in Figure 2. Its 3D positioning is explained below.

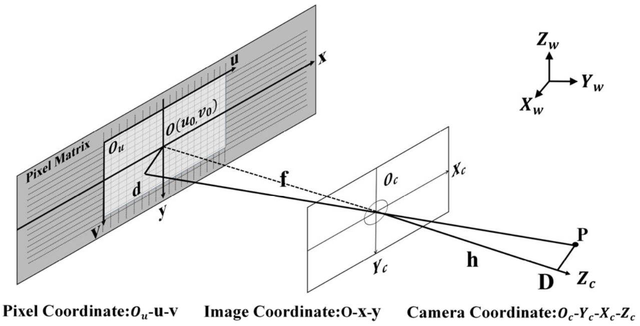


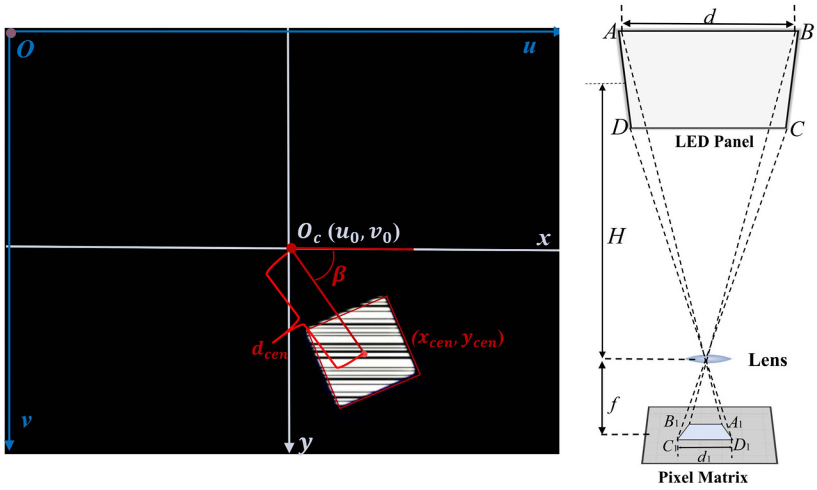
Figure 2. Imaging principle.

Figure 3a is an image of visible light communication which contains much positional information. The pixel coordinates (u_{cen}, v_{cen}) of the center of the light mass can be obtained directly in the image. Its image coordinates (x_{cen}, y_{cen}) can be obtained by Equation (3).

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{3}$$

Then, the length of d_{cen} can be obtained by the calculation of Equation (4).

$$d_{cen} = \sqrt{x_{cen}^2 + y_{cen}^2} \tag{4}$$



(**a**) (**b**)

Figure 3. Original positioning principle; (**a**) is the principle of 2D positioning, (**b**) is the principle of height calculation.

According to the image coordinates of the light, the pinch angle β in Figure 3a can be calculated. After the length d_{cen} and the angle β are known, the 2D coordinate of the light relative to the camera is obtained utilizing the imaging relation in Figure 2.

Figure 3b illustrates a conventional method of height calculation. The four outer vertices $A(x_1, y_1, z_1)$, $B(x_2, y_2, z_2)$, $C(x_3, y_3, z_3)$, and $D(x_4, y_4, z_4)$ of a rectangular light lie in a plane with equal Z coordinates. The corresponding projected points of the four vertices in the image are $A_1(x_{11}, y_{11})$, $B_1(x_{12}, y_{12})$, $C_1(x_{13}, y_{13})$, and $D1(x_{14}, y_{14})$. According to the monocular camera imaging geometry, the vertical distance H between the rectangular light and the camera lens can be calculated using Equation (5):

$$H = \frac{fd}{\max(d_i)}, (i = 1, 2, 3, 4) \tag{5}$$

where d_i is the edge length of the rectangular light in the image. The camera deflection angle causes the geometric projection of the rectangular light to deform, so the maximum d_i is chosen to minimize the error.

From the above description, it is clear that 3D positioning functionality can be achieved using a camera and a rectangular LED. In Section 4 of the experiment, we conducted a comparative analysis between this original algorithm and the proposed algorithm, providing a comprehensive evaluation.

2.3. Issue Analysis

The above positioning method is implemented under the ideal condition where the receiver remains level with the light, as shown in Part 2 of Figure 4. The imaging position at this point correctly reflects the relative positions of camera and light. However, when an angular deflection of the receiver occurs, the imaging of the light in the image undergoes a positional shift and deformation. This results in positioning errors, as shown in Figure 4, Part 1 and Part 3. The receiver angle deviation can have an impact on the 2D position and altitude calculation, which in turn leads to errors in 3D positioning. This is detailed and analyzed below.

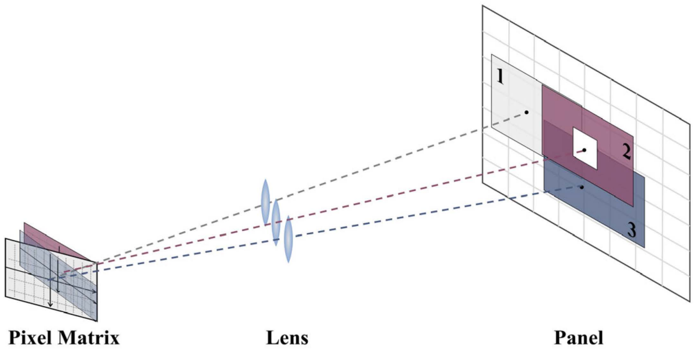


Figure 4. Light position information in the pixel matrix; 1 is left deflection, 2 is normal, 3 is down deflection.

2.3.1. D Error

In order to analyze the specific effects, three scenarios are constructed in Figure 5. The position of the LED in the camera coordinate system reflects the relative position with respect to the camera. Therefore, analyzing the position error of the LED in the camera coordinate system reflects the position error of the camera.

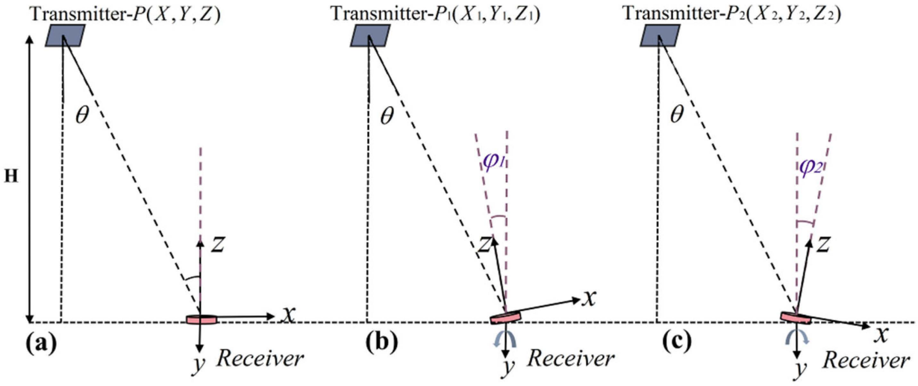


Figure 5. Coordinate change due to angle change; (**a**) is initial scene, (**b**,**c**) are after rotation.

Figure 5a shows that the initial coordinates of the LED in the camera coordinate system are $P(X,Y,Z)$ without any rotation around the Y-axis. The LED coordinates could be various because of the camera's rotation angle. Figure 5b,c shows two rotation scenarios in different directions. From Figure 5b to Figure 5a, the coordinates of the camera coordinate system of the LED are changed from $P_1(X_1,Y_1,Z_1)$ to $P(X,Y,Z)$. The mathematical equation is expressed as follows.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R_Y(\varphi_1)\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = \begin{bmatrix} \cos\varphi_1 & 0 & \sin\varphi_1 \\ 0 & 1 & 0 \\ -\sin\varphi_1 & 0 & \cos\varphi_1 \end{bmatrix}^{-1}\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} \tag{6}$$

After the camera is rotated, the change in height between it and the LED is so slight that it can be ignored. Therefore, the 2D coordinate error obtained before and after camera rotation can be expressed as follows.

$$Eerror = \sqrt{(X_1 - X)^2 + (Y_1 - Y)^2} \tag{7}$$

2.3.2. Height Error

When the angle of the receiver is changed, both perspective transformations and affine transformations result in the distortion of the light image. When projecting a rectangle in three-dimensional space onto a two-dimensional image plane, the lengths, angles, and coordinate positions of the edges change due to rotation and scaling.

Combining the 2D error with the height error, we can mathematically represent the 3D positioning error as (8).

$$Eerror = \sqrt{(X_1 - X)^2 + (Y_1 - Y)^2 + (Z_1 - Z)} \tag{8}$$

2.4. Solution Concept

Above, we describe the basic optical imaging positioning algorithm and the effect of the angle on its results. The implementation of these original algorithms is based on imaging principles, which only use the modulated LEDs as beacons for assisted positioning, and do not really incorporate the characteristics of optical camera communication.

For this reason, we started with the signal frames and found properties that allow for positioning. As shown in Figure 6b, we increase the brightness and exposure of the original frame. It was found that the light signal stripes were still present outside the area of the LED, just hard to distinguish with the naked eye. The light intensity of the stripes in the picture is diffusely attenuated, with different attenuation characteristics at different positions. The light intensity weakening in the picture is reflected in the gray value of the reduction, based on the characteristics of the change in grayscale for positioning. As shown in Figure 6c, we change the camera pose and the imaging areas in the signal frame, which undergo deformation. Through the principles of perspective transformation and affine transformation, it is known that different deformations represent different camera poses and are regular.

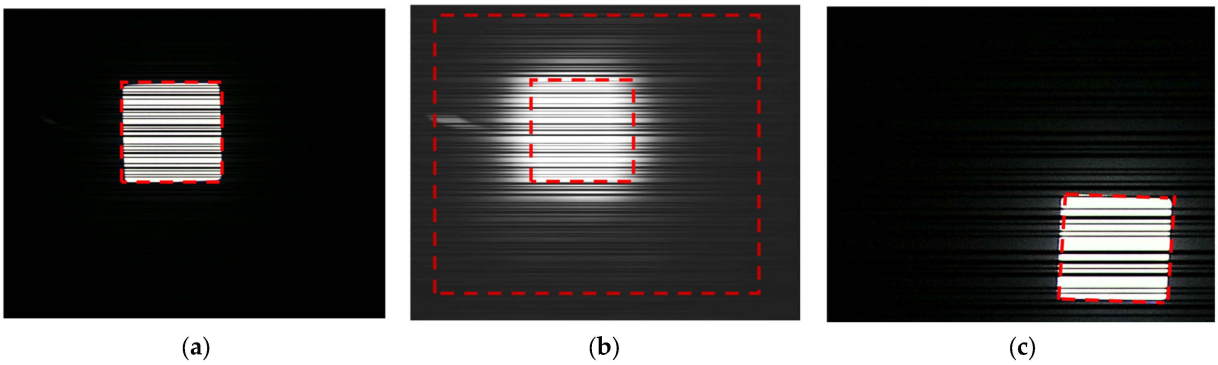


Figure 6. Images of different parameters; (**a**) is the original image, (**b**) brightness up by 20, exposure up by 5, (**c**) is rotated at an angle of 10.

Combining these findings, we consider fusing two features, light intensity and imaging distortion, to achieve positioning that can cope with angular variations. These features are difficult to extract using conventional image-processing algorithms and the workload is enormous. For this reason, we consider extracting these features using a convolutional neural network and propose the MHA-Resnet50 model.

3. Advanced MHA-Resnet50 Model

As shown in Figure 7, the backbone network of the model is Resnet50, which incorporates a multi-head attention mechanism in the middle. The input to MHA-Resnet50 is the signal frames from the camera at multiple angles and coordinates. After extracting features by multilayer convolution, a regressor is used to predict the 3D coordinates of the camera. Its implementation is described in detail below.

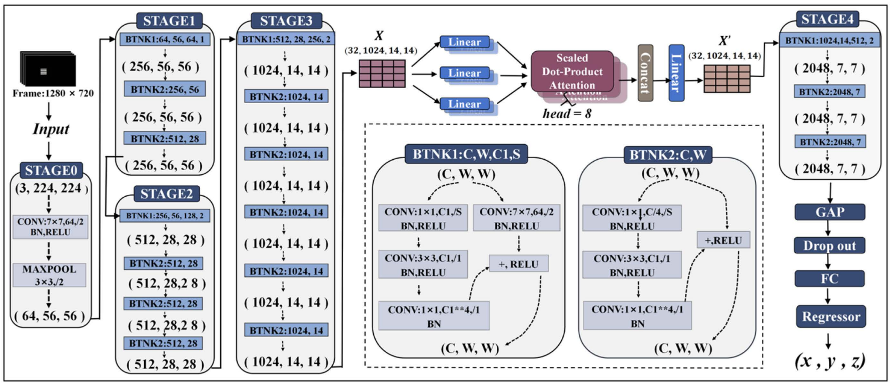


Figure 7. MHA-Resnet50 model structures.

Initially, signal frames with a resolution of 1280 × 720 are normalized, activated, and subjected to max-pooling operations.

In STAGE 1, the model primarily extracts low-level features, with feature map sizes large enough to capture rich spatial detail information.

In STAGE 2, the model reduces the spatial size of the feature maps while increasing the number of channels. This enables the identification of more complex shape features in the light signal regions of the frames.

In STAGE 3, the model further increases the number of channels and compresses the spatial size of the feature maps.

After STAGE 3, the network has formed a rich set of abstract features. Introducing the multiple attention mechanism in this stage can effectively highlight the light intensity change features by assigning weights. It helps the model to better understand the key information in the image.

In STAGE4, the feature maps are then subjected to a convolution operation. A global average pooling process is then performed to reduce the number of model parameters. To further enhance the generalization ability of the model and reduce the risk of overfitting, we add the dropout module. It can randomly drop some features to reduce the model's dependence on specific features [21].

Next, the feature vectors are linearly processed by the FC layer and passed to the regressor xgboost, which finally predicts the 3D coordinates.

3.1. Resnet 50 Model

ResNet50 is a classic convolutional neural network structure in the field of deep learning belonging to the category of residual networks. ResNet50 introduces the residual connection mechanism, which establishes direct connections between different layers. This allows features learned in shallower layers to be passed directly to deeper layers. Consequently, the gradient can still propagate efficiently even as the network becomes deeper. Thus, the problem of vanishing gradient is avoided and the stable training of deep neural networks is ensured.

3.2. Multi-Head Attention Mechanism

This part provides the motivation for using the multi-attention mechanism in the RenNet50 backbone network through test results and details how the mechanism reduces the risk of overfitting and improves the training efficiency.

3.2.1. Motivation

As shown in Figure 8a, when ResNet50 is used to process signal frames to extract features, its convolution operation uses a convolution kernel to slide over the image and compute a weighted sum of the partial area to extract features. As the convolutional layers are stacked layer by layer, high-level features are gradually extracted from the low-level features. In Section 4.2.4, we used the ResNet50 model for training and extracted features from the convolutional layer to draw feature maps and heat maps. The results show that the model only extracts the deformation features of the imaging region and fails to extract the change features of the light intensity. In order to extract both features simultaneously, we use the attention mechanism.

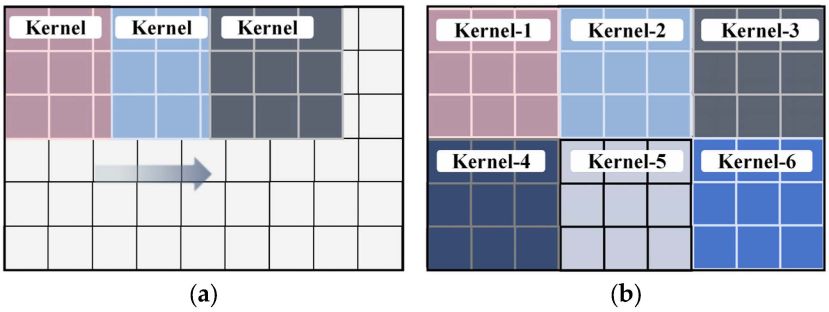


Figure 8. Convolution; (**a**) is ResNet50, (**b**) is MHA-ResNet50.

Attentional mechanisms simulate the perceptions of cognitive functions that are integral to humans. An important characteristic of perception is that humans do not process all information immediately. Instead, we selectively focus on a portion of the information when and where it is needed. Meanwhile, other perceptible information is ignored [22]. This mechanism can help the neural network to process the input data more efficiently. It can distribute different attentional weights between different positions or different features. Consequently, it improves the model's ability to perceive and understand the input data and its representation [23].

As shown in Figure 6, a large number of light intensity features are not obvious without manually adjusting the image parameters. Therefore, we need to assign higher weights to light intensity features during feature extraction. Meanwhile, as shown in Figure 6b, the feature area of the signal black and white stripes is large. If we want to fuse the deformation features and the light intensity features of the stripes to jointly characterize the position information, we need to extract these features at the same time and enhance the model's understanding of the dependency between the long-range features. For this purpose, we use the multi-head attention mechanism. As shown in Figure 8b, the multi-head attention mechanism uses multiple convolution kernels simultaneously to acquire features from different positions of the image while performing convolution. It will adaptively assign attentional weights based on the input features, and it can assign higher weights for features with insignificant light intensity variations. In Section 4.2.4, we used the MHA-ResNet50 model for training and extracting features from the convolutional layer to draw feature maps and heat maps. The results showed that a large number of light intensity features in the signal frames were extracted.

3.2.2. Multi-Head Attention Principle

In this research, we use the 8-head attention mechanism, the principle of which is shown in Figure 9.

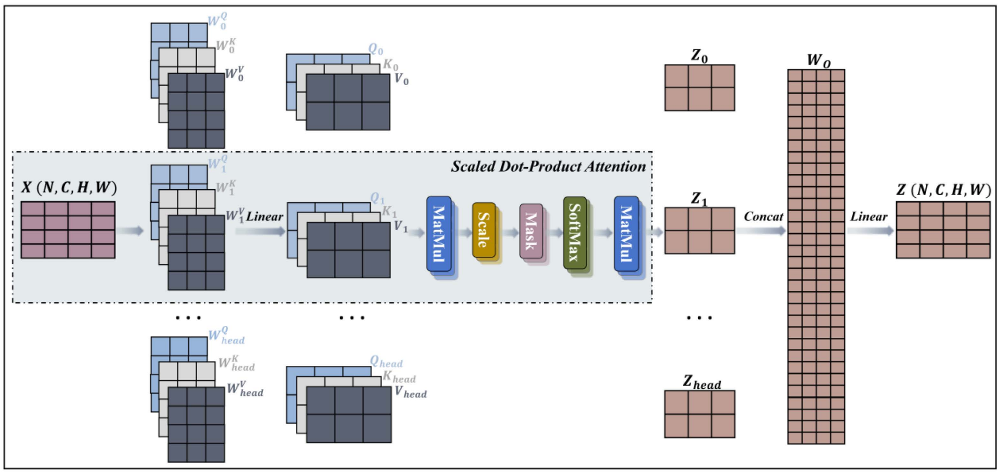


Figure 9. Multi-head attention principle.

The essence of a multi-head attention mechanism is indeed the combination of multiple self-attention mechanisms. Each attention head independently computes self-attention, capturing different features and contextual information within the input sequence. Through concatenation and linear transformation, these information streams are integrated, thereby enhancing the model's expressive power and performance. In each self-attention mechanism, there exists a query matrix Q, a key matrix K, and a value matrix V. The specific implementation of the self-attention mechanism is the Scaled Dot Product Attention (SDA). Its input is a four-dimensional tensor $X(N, C, H, W)$. X is linearly transformed to Q, K, and V, respectively, by Equation (8):

$$Q = XW^Q, K = XW^K, V = XW^V \tag{9}$$

where W^Q, W^K, and W^V are the weight matrices of queries, keys, and values with dimensions $C \times dk$, $C \times d_k$, and $C \times d_v$. Subsequently, the dot product between the query and the key is computed and divided by a scaling factor $\sqrt{d_k}$ to prevent the value from being too large.

$$A = \frac{QK^T}{\sqrt{d_k}} \tag{10}$$

The dot product results are normalized using softmax to obtain the attention weight matrix.

$$S = soft\max(A) \tag{11}$$

The summation is weighted to obtain the expression for attention.

$$Attention = SV \tag{12}$$

Then, the SDA is collapsed and expressed as follows.

$$Attention(Q, K, V) = soft\max(\frac{QK^T}{\sqrt{d_k}})V \tag{13}$$

The concept of the multi-head attention mechanism is to employ the different parameters W^Q, W^K, and W^V to successively perform linear transformations on the matrices Q, K, and V. The results of these linear transformations are then input into the SDA. The computation result is denoted as $head_i$, and its expression is given by the following.

$$head_i = Attention(Q_i, K_i, V_i), i = 1, \ldots, h \tag{14}$$

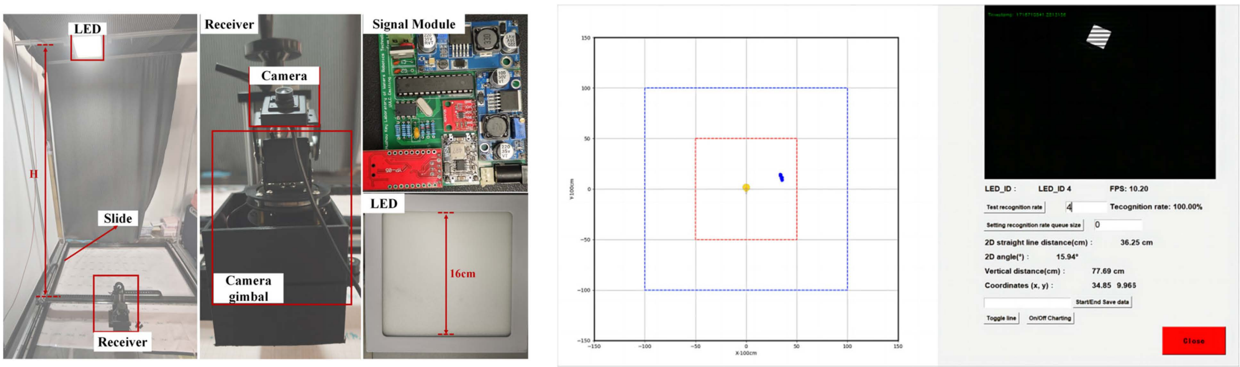
The computed $head_i$ are concatenated into a matrix. It is transformed linearly with the matrix Wo to convert the output of the multi-head attention into a four-dimensional tensor Z. The mathematical representation is as in (17), where h is the number of heads.

$$Z = Mutilhead(Q, K, V) = Concat[head_1; head_2; \ldots; head_h]Wo \tag{15}$$

4. Experiment and Discussion

4.1. Laboratory Testbed

Our algorithm has been tested in an experimental environment. As shown in Figure 10a, all experiments were conducted in an indoor enclosed area of 2.6 m × 2.6 m × 2.2 m. The ground test area was 1 m × 1 m, divided equally into a 10 cm × 10 cm grid with a total of 121 test points. As shown in Figure 10b, to improve the efficiency of the test, we developed the operator interface for the VLP test using Python. The interface allows for the real-time intuitive monitoring of parameter changes during LED recognition and positioning.



LED_ID : LED_ID 4

FPS: 10.20

4

Tecognition rate: 100.00%

0

2D straight line distance(cm) : 36.25 cm

2D angle(°) : 15.94°

Vertical distance(cm) : 77.69 cm

Coordinates (x, y) : 34.85 9.965

(**a**) (**b**)

Figure 10. Testbed; (**a**) is testbed, (**b**) is operator interface.

The transmitter device of the testbed consists of a rectangular LED and a signal modulation module, and is mounted parallel to the ground above the center of the test area. The signal modulation module consists of several off-the-shelf modules, specifically expressed as follows: the microcontroller (MCU) compiles the binary modulation signal into the digital-to-analog conversion module (DAC) after the DAC processing output analog signal. This signal is then fed into the inverting input of the operational amplifier. The MCU inputs a DC bias voltage to the DAC and then inputs the DC bias to the positive phase input of the operational amplifier. The modulated signal and the DC bias are coupled and amplified by the operational amplifier, ensuring that the signal voltage amplitude reaches the operating voltage of the LED. An on-off keying (OOK) modulation scheme is employed to generate control signals for modulating the LED. The detailed parameters of the LED and the modules are shown in Table 1.

Table 1. Parameters of transmitter circuit.

Controller	DAC Module	Power/W	Voltage/V	Dimension/MM
ATmega328P-PU	MCP4725	10	12	160 × 160
Boost Module	**Buck Module**	**NMOSFET**	**Operational Amplifier**	
LM2587	LM2596SDC	TRFB4110	OPA551	

The camera at the receiver was connected to the computer. Before conducting the test, we calibrated the camera so that it could correctly identify the correct position of the LED in the image. Its detailed parameters are shown in Table 2.

Table 2. Parameters of camera.

Resolution	**Pixel Size**	**Image Sensor**	**Format**	**F**
1280 × 720	2.9 µm × 3.0 µm	IMX335	JPG	2.2 mm
Shutter	**Brightness**	**Contrast**	**Gamma value**	**Image gain**
1/2.8CMOS	26	61	500	128

4.2. Model Training Results and Comparative Analysis

4.2.1. Model Parameter Setting

In this study, we built the basic software environment using Torch 2.0.1, Python 3.9, Torch-vision 0.13, and Torch-audio 0.12. The model training was performed on an NVIDIA RTX 4090 graphics processing unit. To ensure consistency and fairness in comparison, the model training parameters were uniformly set. Specifically, we set the batch size to 32 and employed the Adam optimizer for all model training processes. The initial learning rate was set to 0.002. All experimental models underwent training for 200 epochs, during which the probability p of the dropout module was set to 0.3.

In the data preprocessing stage, two image enhancement operations were used to improve the performance and robustness of the model. First, a normalization operation was used to normalize the image data to a range with a mean of [0.485, 0.456, 0.406] and a standard deviation of [0.229, 0.224, 0.225]. This ensures that the data distribution is close to zero mean and unit variance, thus accelerating the convergence of model training. Next, we employed a color dithering operation to increase the diversity of the data. This was achieved by randomly adjusting the brightness, contrast, saturation, and hue of the image. These adjustments improve the model's ability to adapt to different lighting conditions, shooting environments, and color distributions. The combination of the two operations can effectively enhance the feature representation of image data.

To evaluate the performance of the MHA-Resnet50 model and assess the magnitude of positioning errors, we utilized the root mean square error (RMSE) as the loss function. It provides a comprehensive measure of prediction error, and a reduction in RMSE can improve the model's prediction accuracy. In this study, its mathematical expression is given as follows:

$$E_{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[\left(x_{true}^{i} - x_{pred}^{i}\right)^{2} + \left(y_{true}^{i} - y_{pred}^{i}\right)^{2} + \left(z_{true}^{i} - z_{pred}^{i}\right)^{2}\right]} \tag{16}$$

where N is the number of samples per point, $(x_{true}^{i}, y_{true}^{i}, z_{true}^{i})$ is the true coordinates of the ith sample, and $(x_{pred}^{i}, y_{pred}^{i}, z_{pred}^{i})$ is the predicted value of the model for the ith sample.

4.2.2. Data Acquisition

We fixed the LED at a height of 160 cm and 180 cm, respectively. The pitch angle α of the camera was fixed to 0 and the roll angle β was set to −15, −10, −5, 0, 5, 10, 15 at each height, respectively. Finally, a total of 14 sets of data were taken, each set of data had 121 collection points and each point captured 50 frames of images with a resolution of 1280 × 720. A database was created by recording the coordinate position of the images and the information of the rotation angle. The label of the ith point is noted as $C\alpha$:

$$C\alpha = (x_{\alpha,i}, y_{\alpha,i}, z_{\alpha,i}) \tag{17}$$

where $i = 1, \ldots, 121$. In the model training process, the first 40 images of each point are taken as the training set, totaling 67,760 images, and the last 10 images are taken as the validation set, totaling 16,940 images.

4.2.3. Training Results and Comparison

In this part, we analyze the loss curves of the MHA-Resnet50 and set up two sets of tests to compare with multiple models.

In the first group of tests, we compared the MHA-Resnet50 model with widely used convolutional neural network architectures including DenseNet121, MobileNetv2, ResNet50, and ResNet101. Based on the RMSE curves in Figure 11a and the MSE and MAE parameters in Table 3, we found that the proposed MHA-Resnet50 has a large advantage over the other four models. The RMSE is reduced by about 14.478 cm, 14.318 cm, 13.559 cm, and 14.855 cm, respectively. To explore the optimization effect of the multi-head attention mechanism, we incorporated the MHA attention mechanism into the four network structures and conducted the second group of comparative tests.

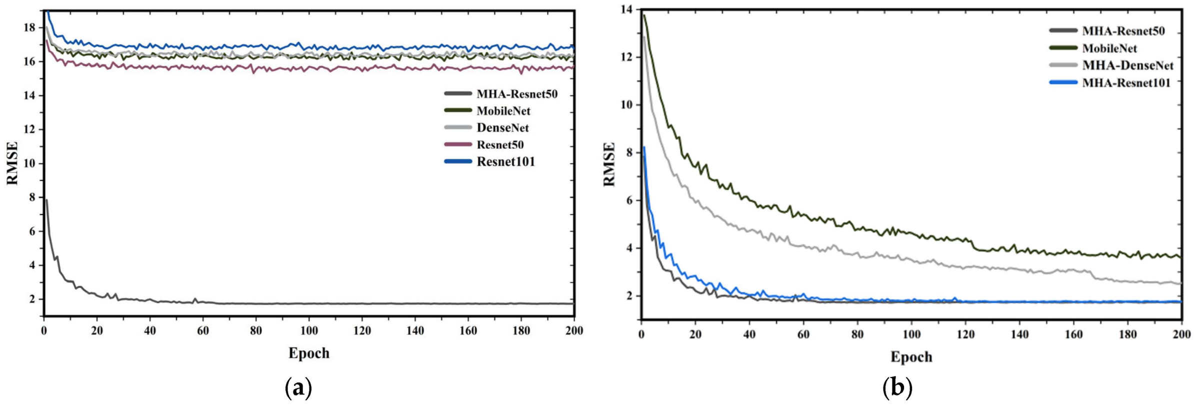


Figure 11. RMSE comparison; (**a**) is the first group, (**b**) is the second group.

Table 3. Model training results of first group.

Model	MSE	RMSE	MAE
DenseNet121	262.41043	16.199087	18.95478
MobileNetv2	257.25336	16.03912	18.806831
Resnet50	233.47543	15.279902	17.386572
Resnet101	274.74146	16.575327	19.563368
MHA-Resnet50	2.960944	1.7207394	0.24662831

The results of the second group of tests are in Figure 10b and Table 4. It can be found that, after incorporating the attentional mechanism, the RMSE of the four models was reduced by 13.696 cm, 12.49 cm, 13.559 cm, and 14.843 cm, respectively. This demonstrates that the addition of the attention mechanism can improve the performance of different models. Moreover, our proposed MHA-Resnet50 exhibits the best coordinate prediction performance.

Table 4. Model training results of second group.

Model	MSE	RMSE	MAE	FLOPs	Params
MHA-DenseNet121	6.2677374	2.503545	1.1392384	2.90GFLOPs	6.96 M
MHA-MobileNetv2	12.592225	3.5485525	2.3558052	326.46MFLOPs	2.23 M
MHA-Resnet50	2.960944	1.7207394	0.24662831	4.13GFLOPs	23.52 M
MHA-Resnet101	3.032443	1.7413912	0.29298633	7.87GFLOPs	42.51 M

4.2.4. Comparison and Analysis of Feature

In this part, we further compare the advantages of using the multi-attention mechanism and verify that MHA-Resnet50 is better than other well-established models.

As shown in Figure 12, we used the eight models trained above to process a frame and plotted the feature map and heat map before and after using the multi-head attention mechanism, respectively. The feature map consists of the superposition of the features of multiple channels in the convolutional layer of the model, which represents the various features captured in the image. The heat map is a further interpretation of the feature map, which visualizes the degree of attention paid by the model to the different regions of the input image. In the feature map and heat map, the color changes from blue to red to indicate that its feature value is gradually getting bigger.

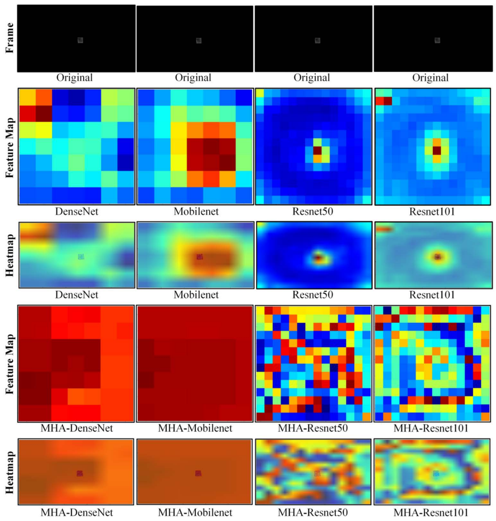


Figure 12. Comparison of feature distributions, color indicates the feature value.

At first, the results of the models were compared before and after the addition of the multi-head attention mechanism. The features of the four well-established models were mostly focused on the imaging region of the LEDs before the addition of the multi-head attention mechanism, and the features of light intensity changes were not extracted. This indicates that the model only focuses on partial features. After the addition, the feature distribution is more balanced. The model extracts a large number of light intensity change features and the imaging region features are also still obvious. This verifies that the multi-attention mechanism pays attention to all regions of the whole image during feature extraction and assigns different weights.

MHA-DenseNet, MHA-MobileNet, and MHA-ResNet were compared after the addition of the multi-attention mechanism. The feature map and heat map colors of MHA-DenseNet and MHA-MobileNet are basically red. This indicates that, although the two models are able to extract features, they have poor importance differentiation ability and can only extract extensive features at a shallower level. The color distribution of MHA-ResNet is more balanced and diverse, which indicates that it can capture both partial details and global features, and the weights are assigned according to importance through the attention mechanism.

Finally, the feature map and heat map of the MHA-ResNet50 and MHA-ResNet101 models were compared. We found that the differences are not significant, and the performance of the two models is close according to the training results in Table 4. For this, we further compared the Floating Point Operations Per Second (FLOPs) and the Params, as shown in Table 4. MHA-ResNet50 clearly has lower requirements on computational resources and memory.

4.3. *Comparative Analysis of Positioning Tests and Results*

In this part, we design a test to evaluate the positioning performance of the model in terms of 2D, height, and 3D.

4.3.1. 2D Positioning Test

In order to test the 2D positioning capability of the proposed algorithm, we compared it with the original algorithm. The camera pitch angle α was fixed to 0 and the LED height was set to 160 cm. For the roll angle β, in addition to the seven angles used for model training, we set six angles of 2.5, −2.5, 7.5, −7.5, 12.5, and −12.5 to test the generalization ability of the model. Similarly, for each angle, 121 test points were taken uniformly in a 1 m × 1 m area. At each point, coordinates were calculated using the original and proposed algorithms, respectively. To minimize the impact of random errors, the results were averaged over 10 consecutive frames for each point.

As shown in Figure 13, we depict the variation curves of the average positioning error of the original and proposed algorithms in 13 sets of tests. When using the original algorithm, the positioning error increases with the angle and the average error is 13.69 cm. In contrast, when using the model for positioning, the error remains stable within the range of 0.82823 cm to 5.73876 cm across all 13 sets. It was not affected by the increase in angle and the average error was 2.185 cm. In addition, for the seven angles used for model training, the average error was 1.114 cm. For the six angles not used for model training, the average error was 4.497 cm. Figure 14 compares the cumulative distribution function (CDF) of the positioning errors of the two algorithms. Figure 15 shows the error distribution, which is relatively uniform. The results show a significant improvement in the error of the modeling algorithm compared to the original algorithm. Overall, the proposed algorithm exhibits a certain degree of generalization capability and resistance to angle variations in 2D positioning.

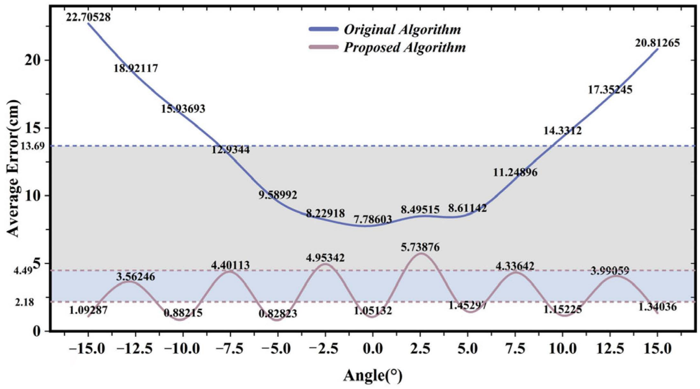


Figure 13. Height 160 cm, average positioning error at different angles.

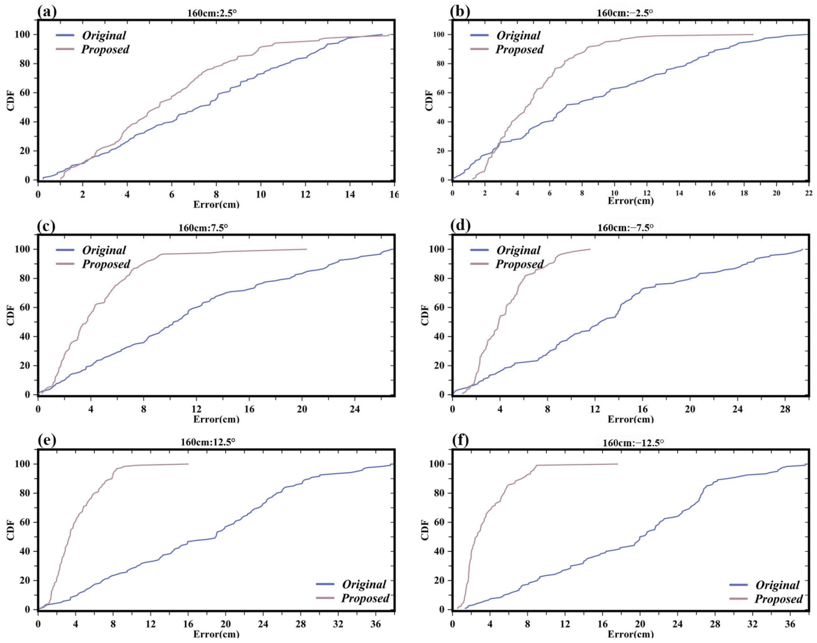


Figure 14. The CDF of 2D positioning errors for different angles; (**a**) 2.5°, (**b**) −2.5°, (**c**) 7.5°, (**d**) −7.5°, (**e**) 12.5°, (**f**) −12.5°.

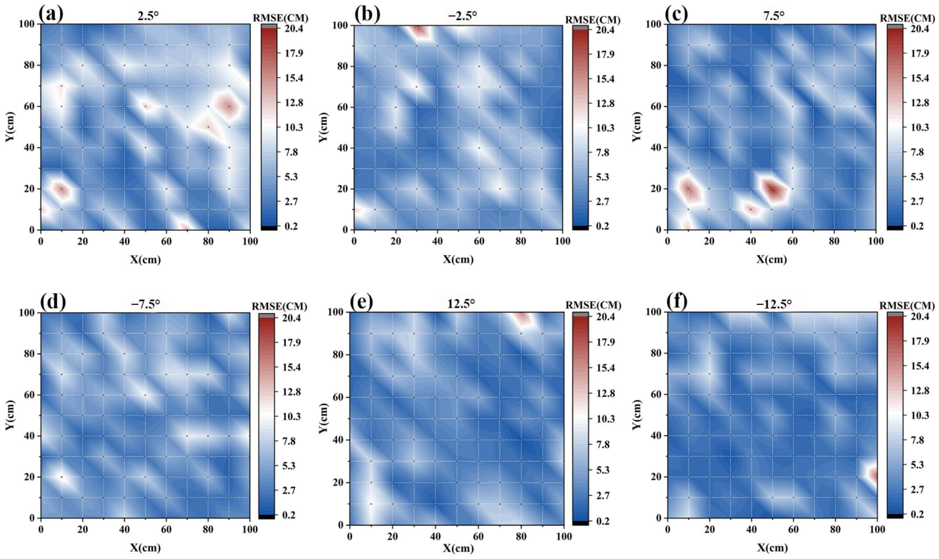


Figure 15. 2D positioning error distribution for different angles; (**a**) 2.5°, (**b**) −2.5°, (**c**) 7.5°, (**d**) −7.5°, (**e**) 12.5°, (**f**) −12.5°.

4.3.2. Height Test

In order to test the capability of the model to predict heights, we similarly conducted a comparison between the original and the proposed algorithm. The camera pitch angle α was fixed at 0°, and the LED heights were set to 160 cm and 170 cm, respectively. Notably, images with a height of 170 cm were not used for model training. For each height, the roll angle β was set to 0 and 12.5, resulting in a total of four sets of tests. Similarly, for each angle, 121 test points were uniformly distributed within a 1 m × 1 m area. At each point, the coordinates were calculated using both the original and the proposed algorithm. To minimize the impact of random errors, the results were averaged over 10 consecutive frames for each point.

Figure 16 shows the error comparison results of the four sets of tests. At a height of 160 cm, as the angle increases from 0 to 12.5, the average error of the original algorithm increases from 4.6 cm to 7.767 cm, while the proposed model algorithm maintains average errors of 1.101 cm and 0.844 cm, respectively. At a height of 170 cm, as the angle increases from 0 to 12.5, and the average error of the original algorithm increases from 9.385 cm to 13.439 cm, whereas the proposed model algorithm achieves average errors of 3.912 cm and 3.736 cm, respectively. According to the results, it can be observed that the error of the original height computation algorithm increases with both the angle and the height. However, with the proposed model algorithm, the error does not increase with the angle. Since the data for the 170 cm height was not used for model training, the error shows a slight increase. In summary, the proposed algorithm exhibits robustness against angle variations and demonstrates a certain degree of model generalization capability in height computation.

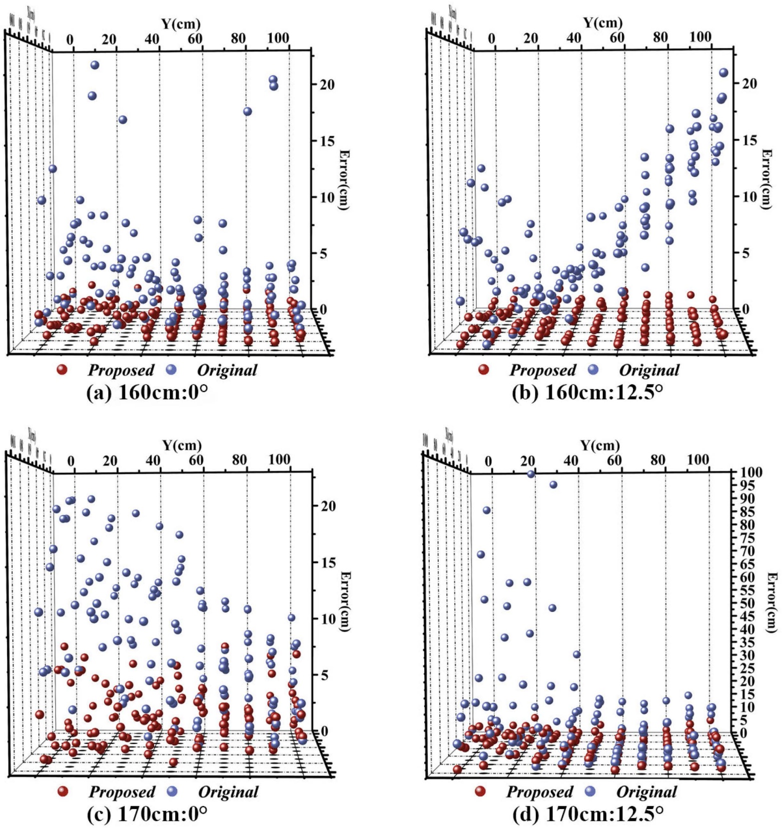


Figure 16. Error comparison at different heights and angles; (**a**) 160 cm:0°, (**b**) 160 cm:12.5°, (**c**) 170 cm:0°, (**d**) 170 cm:12.5°.

4.3.3. 3D Positioning Test

In this part, we tested and compared the 3D positioning capability of the model with the same experimental environment settings as during the height test. The CDF of 3D errors for the four sets of tests is shown in Figure 17.

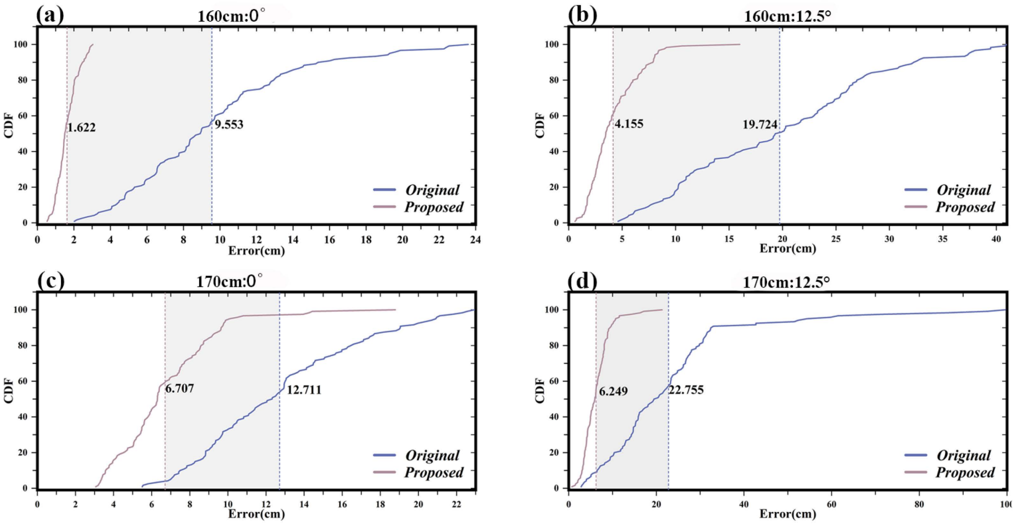


Figure 17. The CDF of 3D positioning errors at different angles and heights; (**a**) 160 cm:0°, (**b**) 160 cm:12.5°, (**c**) 170 cm:0°, (**d**)170 cm:12.5°.

Firstly, considering the test results at a height of 160 cm, as the angle changes from 0 degrees to 12.5 degrees. The 3D average error using the original algorithm increases from 9.553 to 19.724 cm, while the 3D average error of the proposed modeling algorithm only increases from 1.622 cm to 4.155 cm. Since the image data with an angle of 12.5 are not involved in the model training, they are combined with the experimental results of 2D positioning above. We can see that the positioning error of 4.155 cm is within a reasonable range.

Looking again at the two sets of tests for the 170 cm height, we should note that we did not use image data with a height of 170 cm and an angle of 12.5 for model training. When the angle is changed from 0 degrees to 12.5, the 3D average error using the original algorithm increases from 12.711 cm to 22.755 cm. While the proposed model algorithm's average error increases only slightly from 6.707 cm to 6.249 cm, the errors are reduced by about 6.004 cm and 16.506 cm compared to the original algorithm, respectively. Since the image data in these two sets of tests were not involved in the model training, the results can correctly reflect the 3D positioning ability of the model. In summary, our proposed modeling algorithm is resistant to angular changes and has some model generalization ability when performing 3D positioning.

4.4. Discussion

As shown in Table 5, our proposed positioning method eliminates the reliance on IMU when addressing positioning errors caused by camera pose variations. Additionally, we used a medium to low-resolution camera and were able to maintain 3D positioning errors within 7 cm. Compared with [2,3,18], although [2] has a high positioning accuracy, it is poorly convincing with a test deflection angle of only five. In addition, our method only uses the camera and has a lower system cost.

Table 5. Comparison with traditional algorithms.

Require LEDs	Angle (°)	Resolution	Receiver Type	RMSE (cm)	Method	System Cost
1	(−5,5)	1280 × 960	Camera + IMU	2.67	[18]	★★★
1	(−40,40)	2048 × 1536	Camera + IMU	10	[2]	★★★
1	0	Unspecified	Camera + IMU	11.2	[3]	★★★
2	(−40,40)	4032 × 3024	Camera	7.9	[8]	★★
1	(−15,15)	1280 × 720	Camera	6.2	Proposed	★★

The number of stars in the table represents the level of power consumption, with more stars indicating higher power consumption. Compared to [8], our advantage is in the low resolution of the image and the use of only one light. Importantly, the proposed algorithm relies on light intensity variations and the imaging deformation feature and still has a greater potential to ensure highly accurate positioning when coping with greater angular deviations.

5. Conclusions

In visible light positioning systems based on a camera and a single light, changes in camera orientation significantly impact positioning accuracy. Currently, most solutions employ IMU sensors for accuracy compensation. To address this issue, this paper proposes an optimization algorithm based on a convolutional neural network with a multi-head attention mechanism capable of predicting position coordinates when the camera undergoes orientation changes. We design multiple sets of experiments to evaluate the ability of the model to perform 2D positioning, height calculation, and 3D positioning in real time. The experimental results demonstrate that, within an angular range of ±15 degrees, using images with a resolution of 1280 × 720, the model achieves a 2D positioning error within 5.738 cm, height error within 3.912 cm, and 3D positioning error within 6.707 cm. Compared to the original algorithm, the positioning accuracy has been significantly improved. Importantly, this reduces the complexity and cost of the system. Although the maximum angle in the experiments was set to 15 degrees, the algorithm predicts positions based on the relationship between angles and image features. Therefore, it has the potential to handle larger angular deviations. In conclusion, the proposed method provides a reference for the development of indoor positioning technology based on visible light.

In our future work, we will add more training data with light intensity interference caused by varying lighting conditions and changes in LED characteristics to enhance the model's robustness. Additionally, we will explore using lower-resolution signal frames to reduce the algorithm's computational complexity and cost. Finally, we will test larger camera pose variations to evaluate the method's limits.

Author Contributions: Conceptualization, W.J. and X.Z.; methodology, W.J. and X.Z.; software, W.J.; validation, W.J.; investigation, W.J. and X.Z.; resources, Z.W., L.H. and X.Z.; data curation, H.C. and W.J; writing—original draft preparation, W.J. and J.L.; writing—review and editing, W.J., X.Z. and L.H.; visualization, W.J. and X.Z.; supervision, X.Z.; project administration, W.J. and L.H.; funding acquisition, Z.W., L.H. and W.J. All authors have read and agreed to the published version of the manuscript.

Funding: Science and Technology Innovation Activity Program for College Students in Zhejiang Province, China, 2024: [Grant No. 2024R430C027]. Huzhou Key Laboratory of Waters Robotics Technology, Huzhou Science and Technology Bureau: [Grant No. 2022-3]. Zhejiang-French Digital Monitoring Lab for Aquatic Resources and Environment, Department of Science and Technology of Zhejiang Province: [Grant No. 2023-70].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chen, J.; Zeng, D.; Yang, C.; Guan, W. High accuracy, 6-DoF simultaneous localization and calibration using visible light positioning. *J. Light Technol.* **2022**, *40*, 7039–7047. [CrossRef]
2. Cheng, H.; Xiao, C.; Ji, Y.; Ni, J.; Wang, T. A single LED visible light positioning system based on geometric features and CMOS camera. *IEEE Photonics Technol.* **2020**, *32*, 1097–1100. [CrossRef]
3. Hao, J.; Chen, J.; Wang, R. Visible light positioning using a single LED luminaire. *IEEE Photonics J.* **2019**, *11*, 1–13. [CrossRef]
4. Lin, B.; Ghassemlooy, Z.; Lin, C.; Tang, X.; Li, Y.; Zhang, S. An indoor visible light positioning system based on optical camera communications. *IEEE Photonics Technol.* **2017**, *29*, 579–582. [CrossRef]
5. Li, Y.; Ghassemlooy, Z.; Tang, X.; Lin, B.; Zhang, Y. A VLC smartphone camera based indoor positioning system. *IEEE Photonics Technol.* **2018**, *30*, 1171–1174. [CrossRef]
6. Kuo, Y.-S.; Pannuto, P.; Hsiao, K.-J.; Dutta, P. Luxapose: Indoor positioning with mobile phones and visible light. In Proceedings of the 20th Annual International Conference on Mobile Computing and Networking, Maui, HI, USA, 7–11 September 2014; pp. 447–458.
7. Rahman, M.S.; Haque, M.M.; Kim, K.-D. Indoor positioning by LED visible light communication and image sensors. *Int. J. Electr. Comput. Eng.* **2011**, *1*, 161. [CrossRef]
8. Yoshino, M.; Haruyama, S.; Nakagawa, M. High-accuracy positioning system using visible LED lights and image sensor. In Proceedings of the 2008 IEEE Radio and Wireless Symposium, Orlando, FL, USA, 22–24 January 2008; pp. 439–442.
9. Xu, J.; Gong, C.; Xu, Z. Experimental indoor visible light positioning systems with centimeter accuracy based on a commercial smartphone camera. *IEEE Photonics J.* **2018**, *10*, 1–17. [CrossRef]
10. Zhang, R.; Zhong, W.-D.; Kemao, Q.; Zhang, S. A single LED positioning system based on circle projection. *IEEE Photonics J.* **2017**, *9*, 1–9. [CrossRef]
11. Li, H.; Huang, H.; Xu, Y.; Wei, Z.; Yuan, S.; Lin, P.; Wu, H.; Lei, W.; Fang, J.; Chen, Z. A fast and high-accuracy real-time visible light positioning system based on single LED lamp with a beacon. *IEEE Photonics J.* **2020**, *12*, 1–12. [CrossRef]
12. Moon, M.-G.; Choi, S.-I.; Park, J.; Kim, J.Y. Indoor positioning system using LED lights and a dual image sensor. *IEEE Photonics J.* **2015**, *19*, 586–591. [CrossRef]
13. Guan, W.; Zhang, X.; Wu, Y.; Xie, Z.; Li, J.; Zheng, J. High precision indoor visible light positioning algorithm based on double LEDs using CMOS image sensor. *Appl. Sci.* **2019**, *9*, 1238. [CrossRef]
14. Zhao, H.; Zhang, X.; Bader, F.; Zhang, Y. Non-Point Visible Light Transmitter Localization based on Monocular Camera. In Proceedings of the 2021 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Chengdu, China, 4–6 August 2021; pp. 1–5.

15. Qin, C.; Zhan, X. VLIP: Tightly coupled visible-light/inertial positioning system to cope with intermittent outage. *IEEE Photonics Technol.* **2018**, *31*, 129–132. [CrossRef]
16. Bera, K.; Parthiban, R.; Karmakar, N. A Truly 3D Visible Light Positioning System using Low Resolution High Speed Camera, LIDAR, and IMU Sensors. *IEEE Access* **2023**, *11*, 98585. [CrossRef]
17. Hou, Y.; Xiao, S.; Bi, M.; Xue, Y.; Pan, W.; Hu, W. Single LED beacon-based 3-D indoor positioning using off-the-shelf devices. *IEEE Photonics J.* **2016**, *8*, 1–11. [CrossRef]
18. Huang, H.; Feng, L.; Ni, G.; Yang, A. Indoor imaging visible light positioning with sampled sparse light source and mobile device. *Chin. Optics* **2016**, *14*, 090602. [CrossRef]
19. Kim, J.Y.; Yang, S.H.; Son, Y.H.; Han, S.K. High-resolution indoor positioning using light emitting diode visible light and camera image sensor. *IEEE Photonics J.* **2016**, *10*, 184–192. [CrossRef]
20. Naseer, T.; Burgard, W. Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In Proceedings of the 2017 IEEE RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 1525–1530.
21. Wang, S.; Manning, C. Fast dropout training. In Proceedings of the International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; pp. 118–126.
22. Philipp, G.; Song, D.; Carbonell, J.G. Gradients Explode-Deep Networks are Shallow-ResNet Explained. In Proceedings of the 6th International Conference on Learning Representations ICLR Workshop Track, Vanvouver, BC, Canada, 30 April–3 May 2018.
23. Guo, M.-H.; Xu, T.-X.; Liu, J.-J.; Liu, Z.-N.; Jiang, P.-T.; Mu, T.-J.; Zhang, S.-H.; Martin, R.R.; Cheng, M.-M.; Hu, S.-M. Attention mechanisms in computer vision: A survey. *Comput. Vis. Media* **2022**, *8*, 331–368. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.